

JIRA-Style Sprint Plan

Customer Onboarding & Workflow Automation Platform

This sprint plan breaks development into **clear, executable JIRA-style sprints** with epics, stories, and outcomes. Each sprint is designed to deliver a demoable increment.

Sprint 0: Planning & Setup

Duration: 1 week

Goal: Establish a solid foundation and avoid rework

Epics

- Project Planning & Architecture

User Stories / Tasks

- Define system architecture and finalize `architecture.png`
- Review and finalize user stories
- Define onboarding statuses and role flow
- Finalize database schema
- Create GitHub repository and initial README

Definition of Done (DoD)

- Architecture diagram committed
 - User stories documented
 - Repo initialized and accessible
-

Sprint 1: Backend Core – Onboarding APIs

Duration: 1 week

Goal: Enable customer onboarding via REST APIs

Epics

- Backend Core Development

User Stories / Tasks

- Initialize Spring Boot project
- Create JPA entities: CustomerOnboarding, ApprovalHistory

- Configure Oracle datasource
- Implement POST /api/onboarding
- Implement GET /api/onboarding/{id}
- Add validation and duplicate checks
- Basic exception handling

DoD

- Customer onboarding works via Postman
 - Data persists correctly in Oracle
 - Validation errors handled cleanly
-

Sprint 2: Approval Workflow & Business Logic

Duration: 1 week

Goal: Implement role-based approvals and status transitions

Epics

- Approval & Business Rules

User Stories / Tasks

- Define approval roles and permissions
- Implement POST /api/onboarding/{id}/action
- Enforce valid status transitions
- Persist approval audit records
- Handle rejection scenarios

DoD

- Ops → Manager → Compliance flow works via API
 - Approval history stored correctly
 - Invalid actions rejected by backend
-

Sprint 3: Appian Integration & Workflow

Duration: 1 week

Goal: Enable end-to-end workflow through Appian

Epics

- Workflow Automation

User Stories / Tasks

- Build Appian onboarding form (SAIL)
- Configure REST connected systems
- Create Appian process model for approvals
- Configure role-based task routing
- Display onboarding status in Appian

DoD

- Full onboarding flow works via Appian UI
 - Backend APIs successfully integrated
 - Role-based task visibility enforced
-

Sprint 4: Audit, Visibility & Documentation

Duration: 1 week

Goal: Make the system production-like and presentable

Epics

- Audit & Observability

User Stories / Tasks

- Display approval history in Appian
- Add logging and global exception handling
- Generate Swagger/OpenAPI documentation
- Improve dashboards and UI clarity

DoD

- Audit trail visible and complete
 - APIs documented via Swagger
 - System ready for demo
-

Sprint 5: Testing, Polish & Demo Readiness

Duration: 1 week

Goal: Finalize project for portfolio and interviews

Epics

- Quality & Delivery

User Stories / Tasks

- Write unit tests for service layer
- Create Postman collection
- Finalize GitHub README
- Add screenshots and diagrams
- Prepare demo walkthrough

DoD

- Tests passing
- Documentation complete
- Project is interview-ready



Notes

- Each sprint produces a demoable increment
- No UI polish before backend logic is stable
- Backend remains source of truth for business rules

End of JIRA-Style Sprint Plan